Int J Adv Manuf Technol (2006) 27: 1058–1070
DOI 10.1007/s00170-004-2325-y

ORIGINAL ARTICLE

C.K. Mok · F.S.Y. Wong

Automatic feature recognition for plastic injection moulded part design

Received: 15 May 2004 / Accepted: 6 July 2004 / Published online: 16 March 2005

Abstract The design of plastic injection moulded part is a highly iterative process and depends largely on the human expertise including product designers, mould designers, manufacturing engineers, who need to possess thorough and broad specific knowledge and experience. The automatic recognition of features for injection moulded parts is the first prominent stage for any automatic design evaluation system. In this paper, features of plastic product are classified in three hierarchical levels of primitive features, complex features and high-level complex features. The proposed method adopts a hybrid approach, which combines the graph theoretic approach with the logic approach. Methodologies of enhanced multi-attributed adjacency graph (EMAAG) and feature-interaction feature graph (FIFG) for automatic extraction of primitive features and complex features of plastic parts, respectively, are presented. The test example shows that the methodologies developed can recognise various features of injection moulded parts successfully.

Keywords Feature interaction · Feature recognition · Plastics injection moulded part design · Plastics part features

1 Introduction

Injection moulded part design is a highly iterative process, which involves a substantial heuristic knowledge component (practical knowledge) about areas of customer requirements, part design specifications, mould design characteristics, mould making methods and production requirements. Product designers are required to possess a high standard of specific knowledge and experience because detailed decisions require knowledge of the interaction between different parameters. A change of an individual factor in a more favourable direction, for instance a product feature could have a negative effect on other critical factors, for instance the mould design and making. Product design does not result from a quantitative analysis but comes within a range of design procedures, which are partly creative. Individual elements of the design may be opened to quantitative analysis but these do not help the designer to establish the overall form of the design.

In conventional practice, the design of an injection moulded part depends largely on the human expertise including product designers, mould designers, manufacturing engineers, who need to possess thorough and broad specific knowledge and experience. Unfortunately, there is a shortage supply of these experienced designers and engineers to cope with the growing demand in the plastics industry. It is thus quite often that many product design changes are necessary during the stages of mould design, mould making and production in order to meet mould manufacturability requirements as well as production economics. These problems inevitably lead to longer lead times and higher costs.

With the aims of improving the efficiency and effectiveness for the development of injection moulded parts, design automation or supporting systems for injection moulded part design are deemed necessary. The automatic recognition of features of injection moulded parts is usually the first prominent stage for any design automation systems. In this research work, features of plastic products are classified in three hierarchical levels of primitive features, complex features and high-level complex features. The proposed method adopts a hybrid approach, which takes advantages of graph theoretic approach and logic approach. Enhanced multi-attributed adjacency graph (EMAAG), which is an extension to disjointed face relationship, has been developed for automatic extraction of primitive features for plastic products. Feature-interaction feature graph (FIFG) is proposed for the recognition of complex features for plastic parts. Based on the methodologies and algorithms developed, a prototype program has been written and implemented successfully with a test example.

C.K. Mok (✉)
Department of Manufacturing Engineering and Engineering Management,
City University of Hong Kong,
Tat Chee Avenue, Kowloon, Hong Kong
E-mail: meckmok@cityu.edu.hk
Fax: +852-27888423

2 Plastic injection moulded part design

A plastic injection moulded part is basically a shell structure with nominal wall thickness between 0.5 mm and 8 mm depending on part size. Different detail design features are overlaid on the surface of a plastic part to serve different functions such as strengthening, location, fastening, etc. In general, the design of a plastic part should meet the requirements of functional/physical attributes, aesthetic attributes, reliability, durability and serviceability [1]. Wood and Ullman [2] have identified the detail design features of an injection moulded part and their related functions as listed in Table 1 and Table 2, respectively, in their study.

Faced with the problem of designing a complex plastic part, it is often helpful to break the design down into its basic design features and to design the individual features correctly. The designed details can then be put back together to create a good quality, economical part which will be easy to produce. The common practice of considering the total part design as one unit often results in a condition where minor, but important, details are overlooked. A minor error in the smallest of details will be found by the melt and may result in part, which is unnecessarily lower in quality or higher in cost. In extreme cases, the faulty part can even result in personal or financial loss.

Based on the above reasons, the present study is confined to detail design features. The authors of this paper have further identified from the work of Wood and Ullman [2] the most common design features of an injection moulded part according to their over 10 years experience in the area of product design and development. These design features are shown in Fig. 1. The current work would focus on these features.

Table 1. Design features of an injection moulded part [2]

Wall
Holes (through and blink)
Ribs/gussets/louvers/bars/grates
Protrusions/projections/tabs/flanges
Bosses/pegs
Grooves/depressions/indents/pockets
Slots
Windows/cutouts
Countersinks
Snap fits
Disks

Table 2. Functions of design features [2]

Provide access	Convey	Link	Slide
Activate	Cover	Mount	Space
Actuate	Create	Orient	Stabilize
Aid	Display	Partition	Strengthen
Align	Eject	Pivot	Support
Allow	Enclose	Position	Transfer
Amplify	Facilitate	Prevent	Transmit
Assist	Fasten	Protect	View
Attach	Guide	Receive	
Avoid	Hide	Reduce	
Conform to	Hold	Repel	
Connect	Join	Restrain	
Constrain	Latch	Rotate	
Contain	Limit	Secure	

3 A review of feature recognition

The problem of automatic feature recognition has attracted a great deal of research efforts from many researchers over the last quarter of a century. Early research work in this area mainly focused on the recognition of single individual machined feature based on B-Rep (boundary representation) method [3, 4]. The boundary representation of the part is compared to the initial work piece and the material to be machined was discretised into laminae. There are various approaches on feature technology, e.g. syntactic pattern recognition method [5–7], logic or expert system approach [8–10], CSG tree manipulation techniques [11], graph theoretic approaches [12–14], etc. Shah [15] has conducted comprehensive reviews of some of these approaches on feature technology.

The graph theoretic approaches concerning feature recognition have attracted a great deal of interest in recent years. In these approaches, both parts and features are represented by graphs comprising of faces, edges, adjacency relationship and attributes such as face attributes and arc attributes. Joshi and Chang [12] defined a collection of heuristics based on the concept of attributed adjacency graph (AAG) in the context of the recognition of polyhedral features such as pockets, slots and holes from a B-Rep model. An AAG is a face-edge attributed graph with faces as nodes and edges as arcs. An attribute (label) is accorded to each arc depending on the nature (concave or convex) of the edge it represents. De Floriani [13] used an algorithm to extract the subgraph components according to the connectivity of the face-edge graph of the part. His work is confined to protrusions and depressions. Chuang [14] used vertex-edge (V-E) graphs to develop his system. Graph-based approach enables the representation of relationships between objects of the same class or level as well as relationships among objects of different levels can be made in an easy manner. Features at different levels can be easily represented by subgraphs. Feature recognition hence becomes a reversal process of subgraph matching.

Wong and Venuvinod [16] adopted a hybrid approach for feature recognition in the context of "design for assembly". They combined the graph-based approach with the logic approach. The method extended the capability of the AAG by introducing the concept of multi-attributed adjacency graph (MAAG) with enhancements on face attributes and arc attributes. The approach extends the applicability of AAG to cylindrical parts. Yuen et al. [17, 18] subsequently extended Wong and Venuvinod's work to cope with the infinite variety of features through a feature coding system. Feature recognition through feature coding is based on the premise that the feature identification process is reduced to the relatively simple task of looking up a finite feature database where all features of interest are listed along with their feature codes.

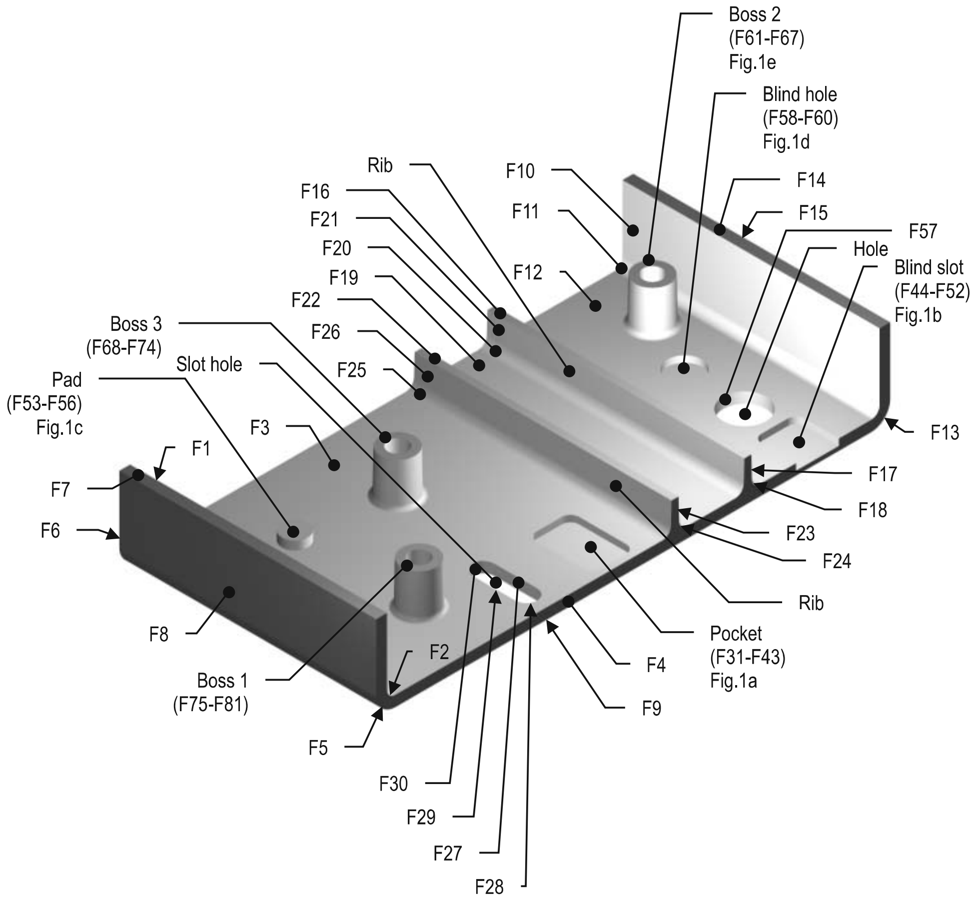


Fig. 1. An injection moulded part with common design features (associated figures in Fig. 16)

The application shape feature recognition approach [19, 20] was purely concentrated on the geometrical aspect with some emphasize in machining. This approach embeds domain specified knowledge that enables a direct, tailor-made and simple approach such as an expert system. Stefano [19] identified six classes of casting features relevant for casting design and core development categories of casting features. Fu [20] defined recognition criteria for undercut features of plastic injection mould. This methodology can be applied to some specific domains. Tate et al. [21] developed a heuristic to extract the symmetric features of a part from a boundary representation solid. Tate's techniques made use of the heuristic of the looped surface properties of the areas of the part to extract the symmetry axes and planes for assembly oriented design application.

Although recognition of part features is important for design evaluation and development of plastic injection moulded parts, there is still little research in this area. Most of the research and development work has been done for the recognition of machined parts. There are significant differences between machined parts and injection moulded parts. A machined part is obtained by removal of material from a stock material using metal cutting processes. Whereas, a plastic injection moulded part is basically a shell structure with different detail design features overlaid on the surface of a plastic part to provide different functions such as strengthening, location, fastening, etc. It is made by the injection moulding process. Hence, there is obviously a potential need to develop a methodology to recognise features of injection moulded parts.

4 The methodology

4.1 Enhanced multi-attributed adjacency graph (EMAAG)

In this study, a methodology called the enhanced multi-attributed adjacency graph (EMAAG) has been developed to facilitate the extraction and recognition for features of plastic products. EMAAG is an enhancement of the MAAG [16] in which both the nodes and arcs can have specified attributes. For instance, in a MAAG, the node attributes may be specified as “pl” for plane, “cyl” for cylindrical, etc., and the arc attributes may be specified as:

1. Arc attribute 0 if the edge is concave ($\theta > 180°$);
2. Arc attribute 1 if the edge is convex ($\theta < 180°$);
3. Arc attribute 2 for an edge with $\theta = 360°$ (within a user-specified limit);
4. Arc attribute 3 for a smooth edge with $\theta = 180°$ (within a user-specified limit) and;
5. Arc attribute 4 for an edge with $\theta = 0°$ (within a user-specified limit).

All the previous work in feature recognition in the context of the graph-based approach were applied to pairs of adjacent faces (i.e. face pair actually intersecting to produce a real physical edge), which is not capable of providing the information for a pair of faces linked up by a fillet face or a chamfer face between them in practical circumstances. In the design of general plastic products, there is always a fillet to separate a pair of faces. For a machined part, sharp edge/sharp corner is normally removed to let a slight chamfer appear between two faces of the actual part. Moreover, a fillet radius is produced at the intersection of two planes due to the existence of the tool nose radius, regardless of the sharpness of the nose of the cutting tool. The adjacency relationship itself is not able to deal with features and product design analysis. It is necessary to extend the notion to pairs of non-adjacent faces as well as disjointed faces. The EMAAG extends the MAAG to include the relationship of the disjointed (non-adjacent) faces. Non-adjacencies are indicated by negative arc attributes as follows:

1. -1 if the pair faces would result in a (virtual) convex edge when suitably extended,
2. -2 if the virtual edge is concave,
3. -3 if the virtual edge is smooth $\theta = 180°$,
4. -4 if the virtual $\theta = 0°$,
5. -5 for a pair of disjointed external parallel faces,
6. -6 for a pair of disjointed internal parallel faces,
7. -7 for a pair of disjointed offset parallel faces.

Thus, EMAAG is capable of representing all the spatial relationships of interest between a pair of faces. In the machining domain, information concerning adjacent faces is of particular importance. In case of plastic product design and product design in other contexts, the main interest lies between pairs of adjacent faces as well as between disjointed faces. EMAAG is capable of supporting the extraction of features of plastic products as well as machined features of plastic injection mould.

4.2 Hierarchical approach to feature recognition for plastic product design

Plastic product features can be classified in classes of three hierarchical levels: i) primitive feature, ii) complex feature, and iii) high-level complex feature. The taxonomy of plastics product features is shown in Fig. 2.

Except for special forms, a plastic part feature composed of special curved surfaces is produced from an instance of any one class of features. The low level features are primitive plastic product features. A primitive feature of a plastic product is composed of two parts, i.e. its primitive root feature and boundary faces. For example, a step feature (see Fig. 3) is represented in a clause that consists of two parameters and two lists:

1. Feature reference number;
2. Feature type, i.e. step;
3. Face list of the root feature, i.e. [2,3,4];
4. List of the boundary faces, i.e. [1,5,6].

The step can thus be represented in the following format:
feature (1, "step", [2,3,4], [1,5,6])

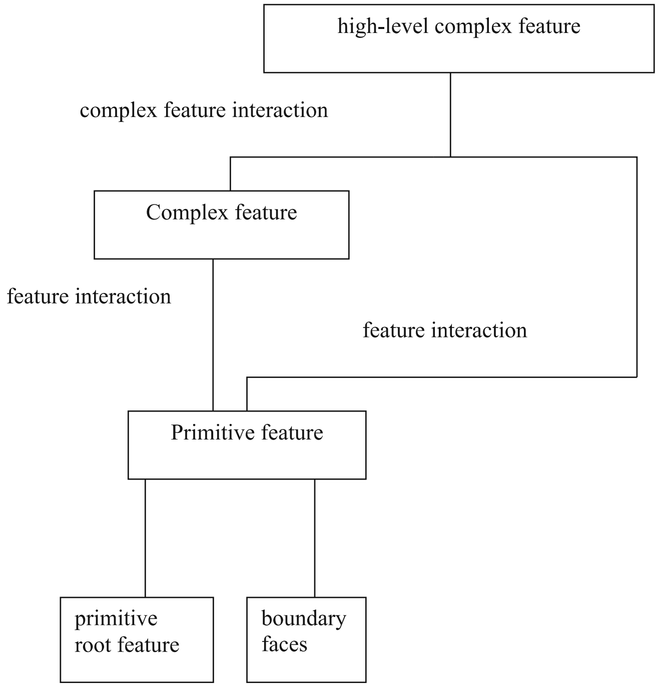


Fig. 2. Taxonomy of plastic product features

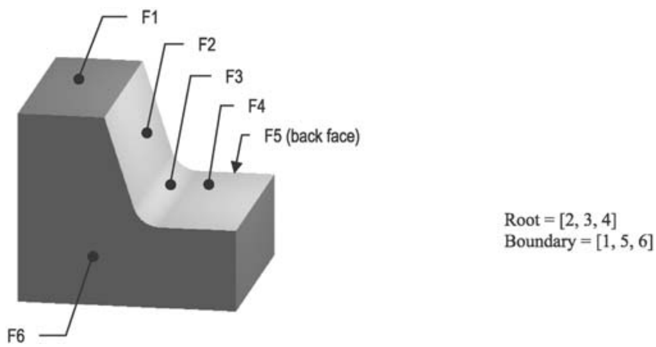


Fig. 3. A step

In this research work five primitive features are identified: step, blind step, pad, hole and blind hole. These five primitive root features, together with their corresponding EMAAG subgraphs are shown in Figs. 4, 5, 6, 7, and 8.

The EMAAG of the subgraphs could be written as production rules in PROLOG as templates for feature representation and recognition. As an example, the rules of a step is as shown below:

Step
/ Root feature of a plastic step*/*
fea("step",[A,B,C]):-
adj(node(A,"plane"),3,node(B,"fillet")),
adj(node(C,"plane"),3,node(B,"fillet")),
adj(node(A,"plane"),-1,node(C,"plane")),
A<>C,
not_a_blind_slot (A,B,C).

A primitive feature is formed by merging the boundary faces of primitive root features. Mutations in members of root fea-

Fig. 4. Step

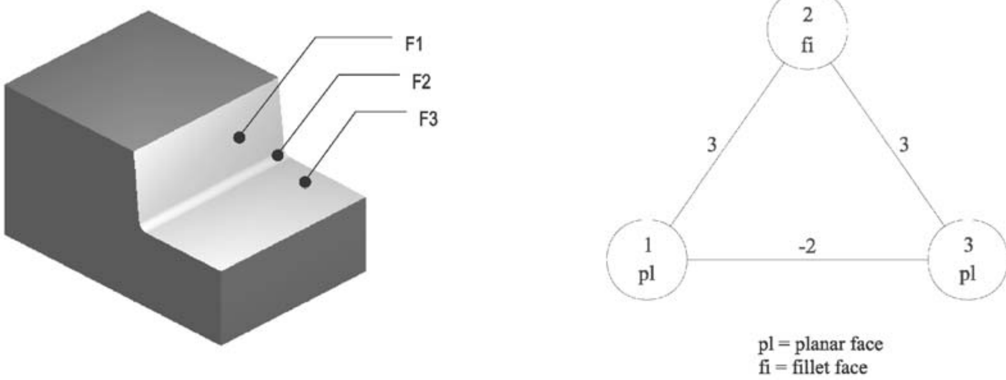


Fig. 5. Blind step

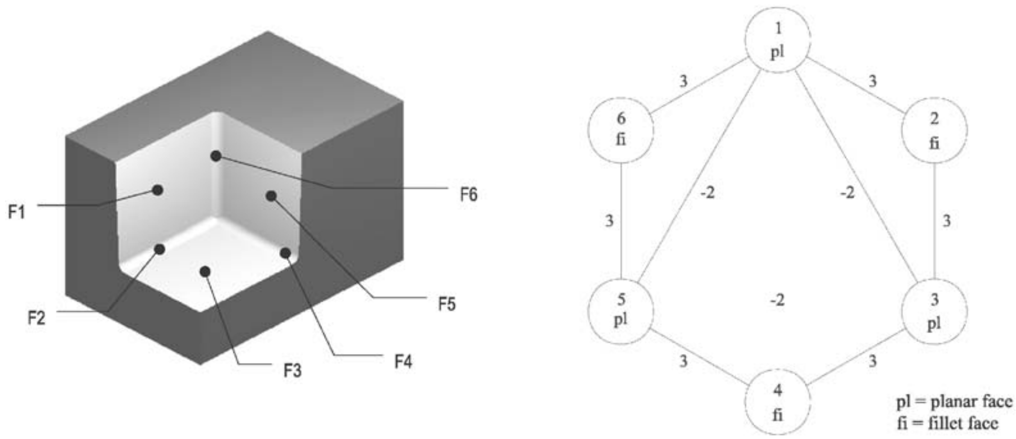


Fig. 6. Pad

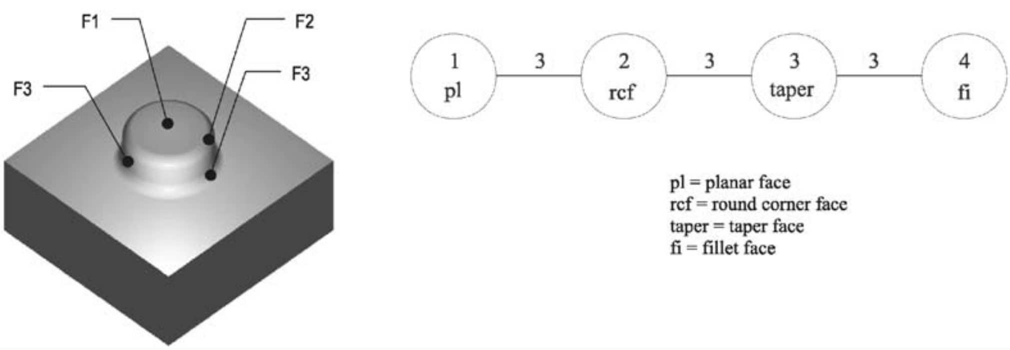
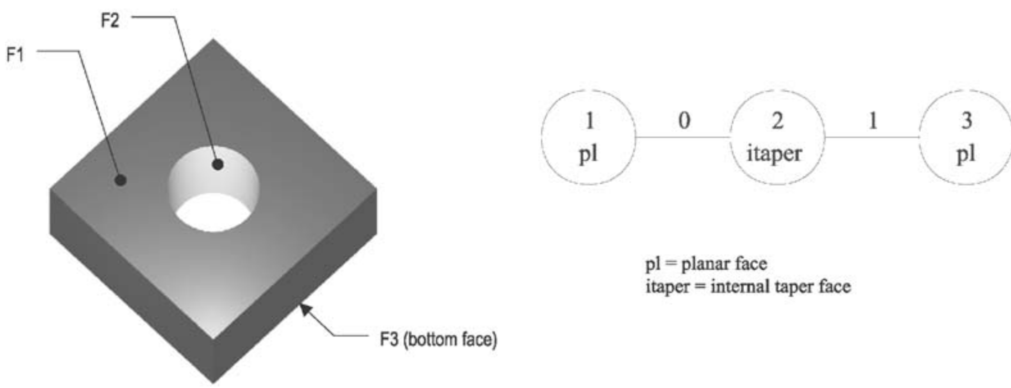


Fig. 7. Hole



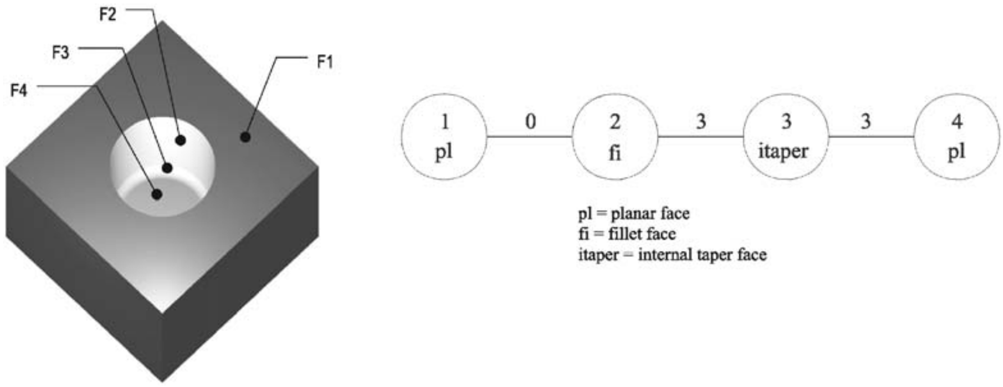


Fig. 8. Blind hole

tures and members of boundary faces would make a family of one primitive feature [16]. A primitive feature may exist in three manners: (i) independently, (ii) interact with another primitive feature to form a complex feature, or (iii) interact with other complex features to form a high-level complex feature. The next level of plastic product features is complex feature, which is formed by interaction of two primitive plastic product features. There are four types of feature interaction: boundary faces to boundary faces (BB) interaction, root faces to boundary faces (RB) interaction, root faces to root faces (RR) interaction and boundary faces to root faces (BR) interaction. In BB interaction, the two features have a common boundary face. In RB interaction, the boundary face of one feature is the root face of the other. In RR interaction, the two features have a common root face. In BR interaction, the root face of one feature is the boundary face of the other.

The highest level of plastic product features is a high-level complex feature, which is formed by one of the following manners:

1. Interaction of a complex feature and primitive features,
2. Interaction of complex features.

Typical high-level plastic product features are boss with 3 ribs, boss with 4 ribs, connected boss, rim, gusset, etc. A typical part with complex features and high-level features is shown in Fig. 9.

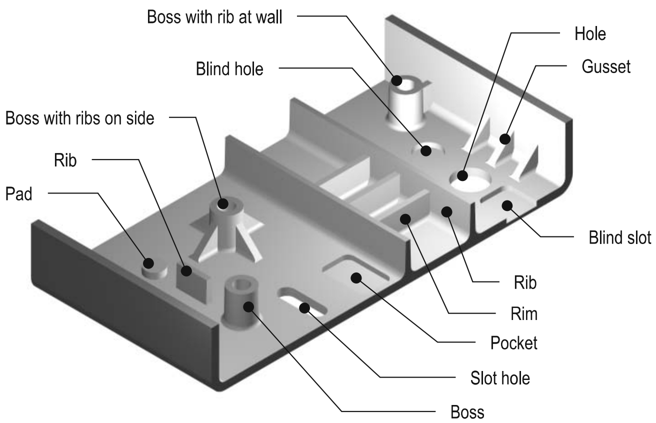


Fig. 9. A typical part with complex features and high level features

4.3 Representation of complex features

4.3.1 Feature-interaction feature graph

A feature-interaction feature graph (FIFG) has been developed in this work to facilitate systematic representation and extraction of a complex feature. The FIFG of a complex feature is a graph $G = (N, A, AL)$, where

- N is a set of nodes representing the N primitive features of a plastic part. The primitive plastic product feature reference number and primitive plastic product feature are used as node labels and node attributes.
- A is a set of arcs representing the feature interaction relationships between the features.
- AL is a set of "feature-interaction" functors containing the parametric values of those feature interaction relationships. The data structure of a "feature-interaction" is as follows:

*/***Data structure of "Feature-interaction" functor***/*
fi*(Type_of_Feature_interaction,Interacted_Face_List).*

The feature-interaction functors for interactions of RR, RB, BR and BB can be written in PROLOG rules for complex feature recognition as shown below:

```
/*Complex Features Recognition*/

/*****frel(ft(Feature_no_of_feature1,Feature1),
fi(Type_of_Feature_interaction_between_feature1 and 2,
 Interacted_Face_List),
ft(Feature_no_of_feature2,Feature2)) ************/

/* Template for "RR" Root-Root Interaction */
frel(ft(N1,F1),fi("RR",I),ft(N2,F2)):-
feature(N1,F1,R1,_),
feature(N2,F2,R2,_),
intersect(R1,R2,I),
N1<>N2.

/* Template for "RB" Root-Boundary Interaction */
frel(ft(N1,F1),fi("RB",I),ft(N2,F2)):-
feature(N1,F1,R1,_),
feature(N2,F2,_,B2),
intersect(R1,B2,I),
N1<>N2.
```

```
/* Template for "BR" Boundary-Root Interaction */
frel(ft(N1,F1),fi("BR",I),ft(N2,F2)):-
feature(N1,F1,_,B1),
feature(N2,F2,R2,_),
intersect(B1,R2,I),
N1<>N2.

/* Template for "BB" Boundary-Boundary Interaction */
frel(ft(N1,F1),fi("BB",I),ft(N2,F2)):-
feature(N1,F1,_,B1),
feature(N2,F2,_,B2),
intersect(B1,B2,I),
N1<>N2.
```

A rib is a typical example of a complex feature obtained by BB feature interactions. A rib is formed by feature interaction of two step features, Step 1 and Step 2, as shown in Figs. 10 and 11.

Fig. 10. Feature interaction of two step features

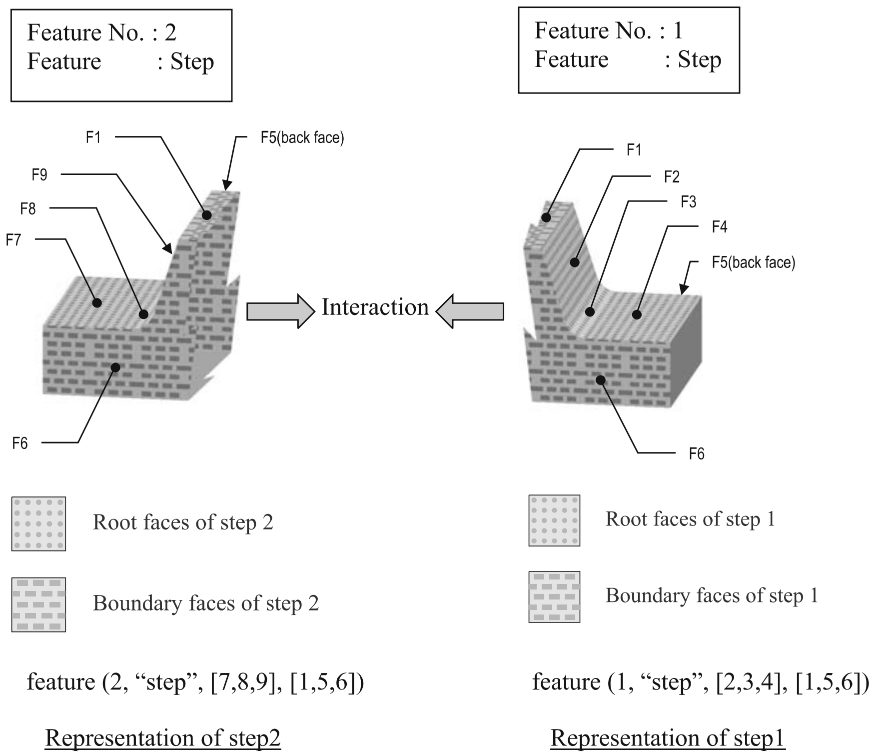
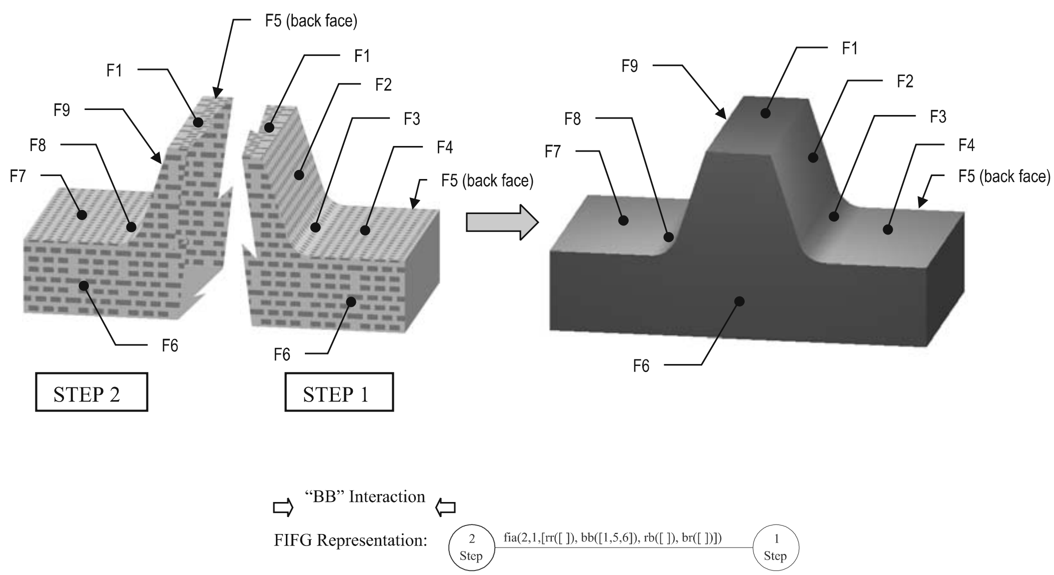


Fig. 11. Feature interaction of two step features to form a rib



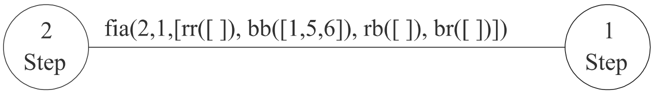


Fig. 12. FIFG of a rib formation

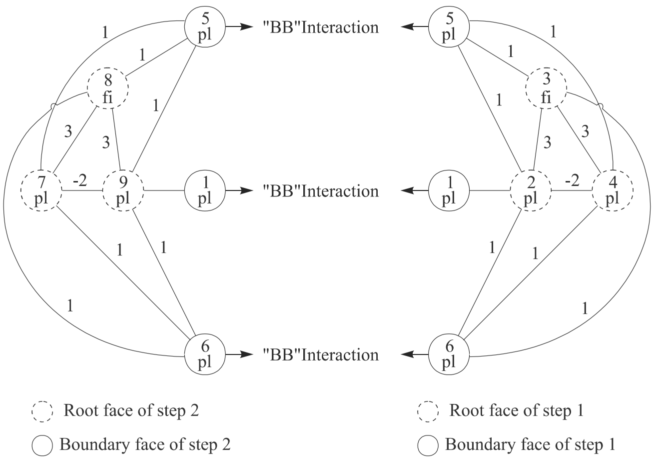


Fig. 13. EMAGG of a rib formation

The FIFG (Fig. 12) and EMAGG (Fig. 13) of a "rib" present the interaction between two primitive features to form a complex feature.

Another example is a slot formed by two steps. Similarly, a blind slot can be obtained by two blind steps through BB and RR feature interactions of two blind steps (Figs. 14a–c). A pocket (Figs. 15a–c) can also be formed from a set of four blind steps. All previous research work defined slot, blind slot, and pocket as primitive features. In this research work, slot, blind slot and pocket can be derived from primitive features and treated as complex features.

4.4 Procedures and algorithms for feature recognition

With reference to the hierarchical structure in Fig. 2, the procedures for plastic product feature recognition are described as follows:

Level 1: Primitive product root feature recognition. The EMAAG of the plastic part can be produced from the data extracted by the B-Rep model of the part. The EMAAG of the plastic part is then inputted into the system. The EMAAG subgraphs of the root primitive features are coded in PROLOG rules as templates for recognition of plastic primitive root features. The outcomes of this level are extracted primitive root features of the plastic part. The extracted primitive product root features are stored in the root feature library.

Level 2: Primitive product feature recognition. A primitive feature is formed by merging the boundary faces of primitive root features. Mutations in members of root features and members of boundary faces would make a family of one primitive feature. This level of recognition is to find the boundary faces of each face of the root feature and the number of the feature. The extracted features are stored in the features database.

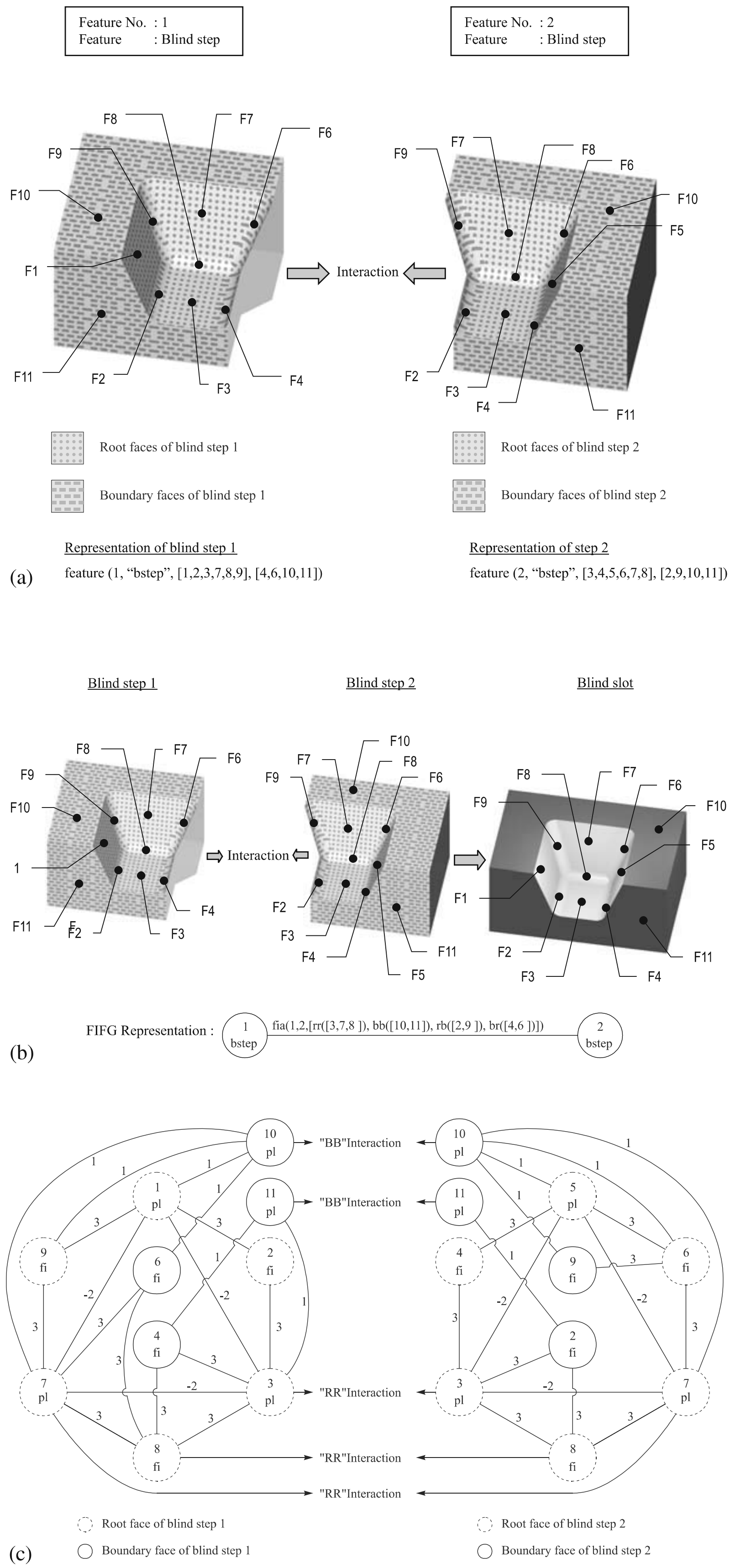
Level 3: Complex product feature recognition. The next level of plastic product features is the complex feature, which is formed by interaction of two primitive product features. The FIFG subgraphs of the four types of feature interactions are coded in PROLOG rules as templates for recognition of complex features.

Level 4: High-level product feature recognition. High-level complex feature may be formed in two manners: (i) interaction of a complex feature and primitive features, and (ii) interaction of complex features. The rules for high-level complex features could also be developed as templates for automatic extraction of high-level complex features.

The present work has been developed up to Level 3. The algorithm for feature recognition up to Level 3 can be written as follows:

/* algorithm of plastics product root feature recognition */
Begin.
/ Level 1: Primitive product root feature recognition */*
Input EMAAG of the part.
Load the part into "EMAAG" database.
Recognise primitive root feature of the part by the matching the first template of the primitive root features.
Assertz the extracted primitive root feature in root feature database.
Repeat primitive root feature extraction by matching with of the templates of primitive root features and assertzing the extracted root primitive feature in root feature database.
Stop root primitive feature recognition at exhaust of templates matching.
Delete all duplicate primitive root features from the root feature database.
End Level 1.
/ Level 2: Primitive product feature recognition */*
Start
Begin feature recognition of first primitive root feature.
Find all faces adjacent to the first face of the first primitive root feature.
Create a boundary face list of the first face.
Repeat finding boundary face and creating boundary face lists for last face of the first primitive root feature.
Merge all face-boundary lists found into one boundary list.
Number the primitive root feature.
Assertz the feature found (with feature no, root feature, boundary list) into features database.
End feature recognition of first primitive root feature.
Repeat feature recognition for all other primitive root feature.
End Level 2.

Fig. 14. **a** Two blind steps to form a blind slot **b** A blind slot formed by two blind steps and corresponding FIFG **c** EMAGG for formation of a blind slot



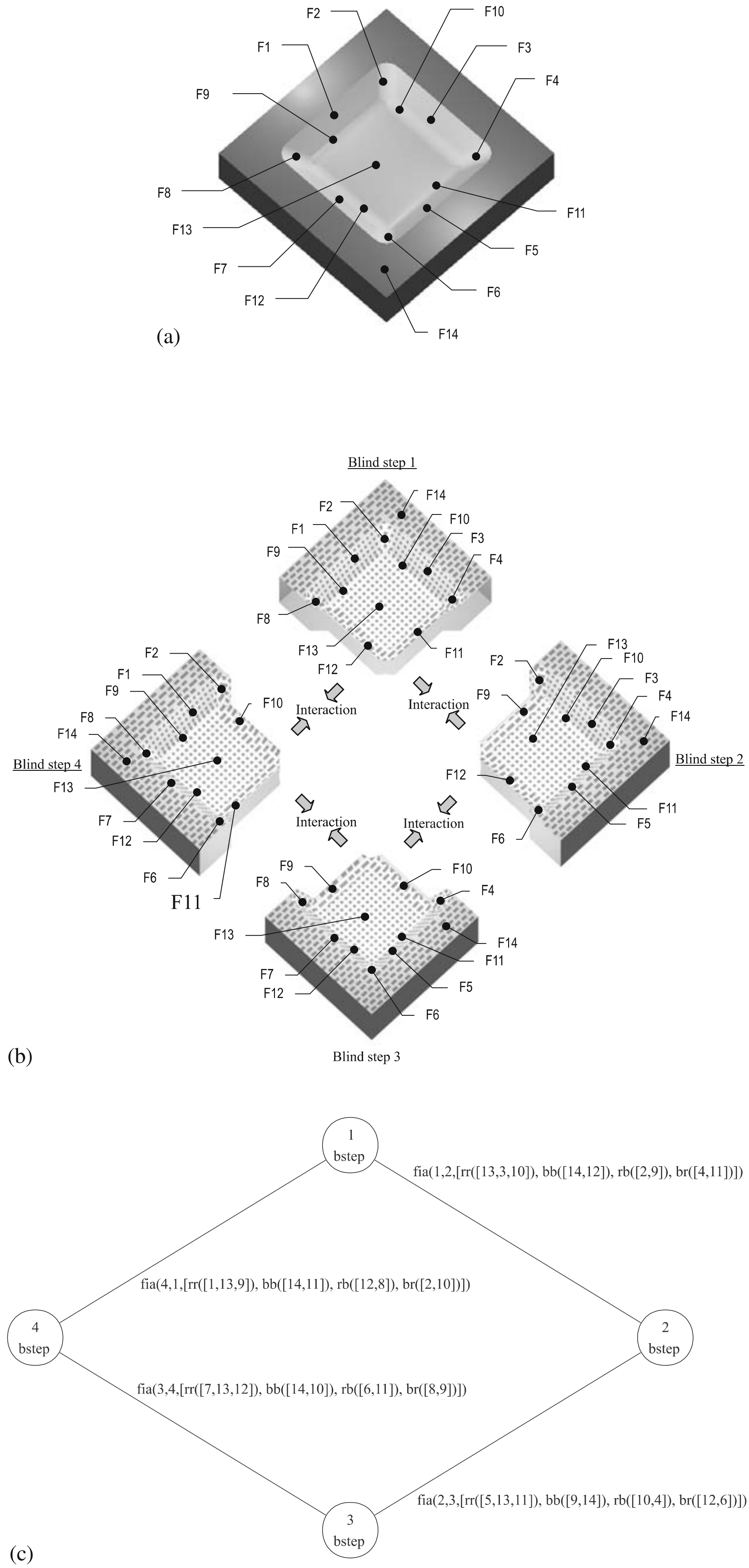


Fig. 15. a Pocket **b** Pocket formed by four blind steps **c** FIFG for formation of a pocket

/ Level 3: Complex product feature recognition */*
Begin
Recognise complex feature of the part by matching the first template of the complex features.
Assertz the extracted complex feature in features database.
Repeat complex feature extraction by matching of the templates of complex features and assertzing the extracted complex features in features database.
Stop complex feature recognition at exhaust of templates matching.
Delete all duplicate complex features from features database.
Number the complex feature in features database.
End Level 3.
End Product Feature Recognition.

5 Implementation and testing

Based on the methodologies and algorithms developed, a prototype program has been written with PROLOG. The plastic product in Fig. 1 (with associated figures shown in Fig. 16) is used as a bench mark to test the program. Firstly, the EMAAG of the part is manually input into the system. The program recognised twenty three plastic features from the second level output which includes 8 step features, 6 blind step features, 2 holes, 3 blind holes and 4 pads.

The recognised features were further processed by the complex plastics feature recogniser. Eight complex features were recognised. All these features have been successfully extracted and recognised as shown in the output listing of the program in Appendix I.

Fig. 16. Associated figures for Fig. 1

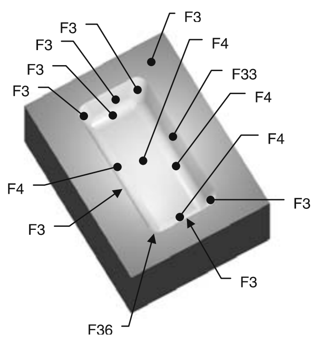


Fig. 1a Pocket

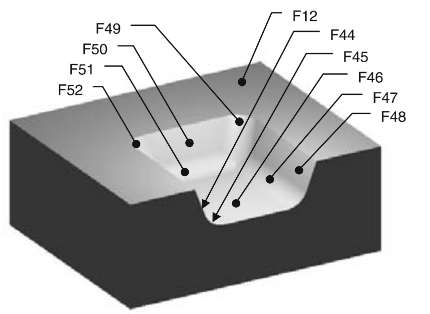


Fig. 1b Blind slot

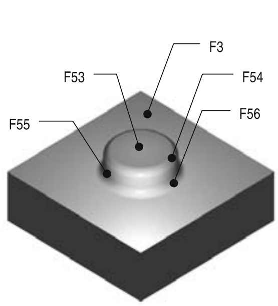


Fig. 1c Pad

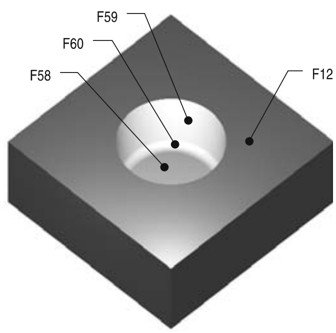


Fig. 1d Blind hole

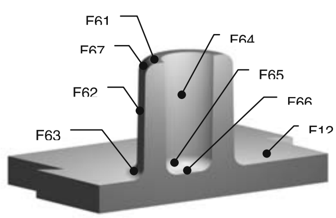


Fig. 1e Boss 2

6 Conclusion and further work

Automatic design evaluation is necessary in order to improve the efficiency and effectiveness for the development of injection moulded parts. The first prominent stage for any design automation systems is the automatic recognition of features of injection moulded parts. In this paper, features of plastics product are classified in three hierarchical levels of primitive features, complex features and high-level complex features. The proposed method adopts a hybrid approach, which takes the advantages of graph theoretic approach and logic approach. Methodologies of enhanced multi-attributed adjacency graph (EMAAG) and feature-interaction feature graph (FIFG) for automatic extraction of primitive features and complex features of plastic parts, respectively, are presented. The methodology has been successfully implemented and tested with a test example. The test example shows that the methodologies developed are efficient in recognition and extraction of various features of moderately complicated injection moulded parts. In order to apply the method for plastic injection moulded parts design, the future direction of this research is along the following two lines:

1. *Development of techniques for high-level features*
   To develop techniques for recognition of high-level features produced by interaction of complex features such as boss with 3 ribs or 4 ribs, connected boss, rims, gusset, etc.
2. *Development of design support system for plastic moulded part design*
   By integrating a design support system with the feature recognition system, the hybrid system could be used for design evaluations and recommendations for the design of plastic injection moulded parts. To automate the whole process, it is necessary to develop an EMAGG generator so that the CAD part data can be converted to EMAAG automatically. Wong [16] had already developed a MAGG generator for automating the input process of MAGG for parts with prismatic and cylindrical features in the AutoCAD environment. An EMAAG generator can similarly be developed in the future.

References

1. Chin KS, Wong TN (1996) Knowledge-based evaluation for the conceptual design development of injection moulding parts. Eng Appl Artif Intell 9(4):359–376
2. Wood SL, Ullman DG (1996) The functions of plastic injection moulding feature. Des Stud 17:210–213
3. Grayer AR (1977) The automatic production of machined components starting from a stored geometry description. In: McPherson D (ed) Advances in computer aided manufacture. North Holland, Amsterdam, pp 137–152
4. Wu MC, Liu CR (1996) Analysis on machined feature recognition techniques based on B-Rep. Comput Aided Des 28(8):603–616
5. Kyprianou LK (1980) Shape classification in computer aided design. Dissertation, University of Cambridge
6. Staley SM, Henderson MR, Anderson DC (1983) Using syntactic pattern recognition to extract feature information from a solid geometric data base. Comput Mech Eng 2:61–66
7. Choi BK, Barash MM, Anderson DC (1984) Automatic recognition of machined surfaces from a 3D solid model. Comput Aided Des 16(2):81–86
8. Henderson MR, Anderson DC (1984) Computer recognition and extraction of form features: a CAD/CAM link. Comput Ind 5:329–339
9. Hirshstick JK, Gossard DC (1986) Geometric reasoning for design advisory systems. Proceedings of the ASME International Computers in Engineering Conference and Exhibition, Chicago, 1986, pp 263–270
10. Dong X, Wozny M (1988) FRAFES, a frame-based feature extraction systems. Proceedings of International Conference on Computer Integrated Manufacturing, Rennselaer Polytechnic Institute, Troy, NY, 23–25 May 1988, pp 296–305
11. Lee YC, Fu KS (1987) Machine understanding of CSG: extraction and unification of manufacturing features. IEEE Comput Graph Appl 7(1):20–32
12. Joshi S, Chang TC (1988) Graph-based heuristics for recognition of machined features from a 3D solid model. Comput Aided Des 20(2):58–66
13. De Floriani L (1987) A graph based approach to object feature recognition. Proceedings of the 3rd Annual Symposium on Computational Geometry, Waterloo, Ontario, Canada, 1987, pp 100–109
14. Chuang SH, Henderson MR (1990) Three-dimensional shape pattern recognition using vertex classification and vertex-edge graphs. Comput Aided Des 22(6):377–387
15. Shah JJ (1991) Assessment of feature technology. Comput Aided Des 23(5):331–341
16. Venuvinod PK, Wong SY (1995) A graph-based expert system approach to geometric feature recognition. J Intell Manuf 6:155–162
17. Yuen CF (1999) Coping with the complexity and infinite variety of geometric features: a cooperative geometric feature recognition system. Dissertation, City University of Hong Kong
18. Yuen CF, Wong SY, Venuvinod PK (2003) Development of a generic computer-aided process planning support system. J Mater Proc Tech 139(1–3):394–401
19. Stefano PD (1997) Automatic extraction of form feature for casting. Comput Aided Des 29(11):761–770
20. Fu MW, Fuh JYH, Nee AYC (1999) Undercut feature recognition in an injection mould design system. Comput Aided Des 31:777–790
21. Tate SJ, Jared GEM (2003) Recognising symmetry in solid models. Comput Aided Des 35:673–692

Appendix I List of output for the test example

feature(1,"pstep",[1,2,3],[7,25,27,28,29,30,31,32,33,34,35,36,37,38,56,74,79,4,6])
feature(2,"pstep",[10,11,12],[14,18,44,52,50,49,48,57,60,67,4,6])
feature(3,"pstep",[17,18,12],[16,11,44,52,50,49,48,57,60,67,4,6])
feature(4,"pstep",[19,20,21],[24,16,4,6])
feature(5,"pstep",[23,24,19],[22,20,4,6])
feature(6,"pstep",[3,25,26],[2,27,28,29,30,31,32,33,34,35,36,37,38,56,74,79,22,4,6])
feature(7,"pstep",[27,28,29],[30,3,9])
feature(8,"pstep",[29,30,27],[28,3,9])
feature(9,"pbstep",[31,43,33,32,39,40],[3,42,38,34,41])
feature(10,"pbstep",[33,35,43,40,34,41],[32,39,3,42,36])
feature(11,"pbstep",[37,35,43,42,36,41],[38,39,3,40,34])
feature(12,"pbstep",[37,31,43,42,38,39],[36,41,3,40,32])
feature(13,"pbstep",[44,50,46,45,52,51],[4,12,47,49])
feature(14,"pbstep",[48,50,46,47,49,51],[4,12,45,52])
feature(15,"phole",[12,57,9],[11,18,44,52,50,49,48,60,67,5,13,

27,28,29,30,4,6])
feature(16,"pbhole",[76,75,80,81],[77])
feature(17,"ppad",[53,54,55,56],[3])
feature(18,"ppad",[64,65,66,67],[63,12])
feature(19,"ppad",[71,72,73,74],[70,3])
feature(20,"ppad",[76,77,78,79],[75,3])
feature(21,"pbhole",[58,59,60,12],[11,18,44,52,50,49,48,57,67,4,6])
feature(22,"pbhole",[61,62,63,64],[65])
feature(23,"pbhole",[68,69,70,71],[72])
complex_feature(1,"boss",features_([16,20]))
complex_feature(2,"boss",features_([22,18]))
complex_feature(3,"boss",features_([23,19]))
complex_feature(4,"rib",features_([3,4]))
complex_feature(5,"rib",features_([5,6]))
complex_feature(6,"oval_slot",features_([7,8]))
complex_feature(7,"pocket",features_([9,10,11,12]))
complex_feature(8,"blind_slot",features_([13,14]))